

*Software Requirements Specification*

*Version 1.0*

*<Date>*

*<Team Name>*

*Lead Software Engineer: <Name>*

*Team Members: <Names>*

*Prepared for*

*MSOE SE2890 Software Engineering Practices*

*Instructor: Walter W. Schilling, Jr.*

*<Term>*

# Table of Contents

Table of Contents ii

1 Introduction 1

1.1 Scope 1

1.2 Purpose 1

1.3 1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

2 Use Case Scenarios 2

2.1 Provide Emergency Warning 2

2.1.1 Actors 2

2.1.2 High Level Description 2

2.1.3 Preconditions 2

2.1.4 Use Case Flow: 2

3 Specific Requirements 3

3.1 User Interface Requirements 3

3.1.1 General requirements 3

3.1.2 Ignition State 3

3.1.3 Grapple Control 3

3.1.4 Direction Control 3

3.1.5 <Write The Rest Here> 3

3.2 Functional Requirements 3

3.2.1 Ignition 3

3.3 External Interfaces 3

3.4 Non-Functional Requirements 4

3.4.1 Performance 4

3.4.2 Reliability 4

3.4.3 Availability 4

3.4.4 Security 4

3.4.5 Maintainability 4

3.4.6 Portability 4

3.4.7 Safety 4

3.4.8 Training-related Requirements 4

3.4.9 Packaging Requirements 4

3.4.10 Legal Requirements 4

# Introduction

## Scope

This specification establishes the functional, performance, and development requirements for Release 1 of a software application for controlling a Dream Cheeky RC Car.

##  Purpose

*The Mindstorm Robot Control Software is an interactive software application that allows the user to operate a Lego Mindstorm Robot remotely from the Windows environment. The Robot is connected to the PC via either a USB connection or a BlueTooth connection. A proxy has been made available for this program which will interface between the robot and the PC..*

## 1.3 Definitions, Acronyms, and Abbreviations

*DI Driver Interface*

*GUI Graphical User Interface*

*SRS Software Requirements Specification*

*TOD Time of Day, typically in reference to a clock.*

##  References

J-STD-016-1995 IEEE/EIA Standard for Information Technology, Software Lifecycle Processes, Software Development, Acquirer-Supplier Agreement

IEEE-STD-P1063 IEEE Standard for Software User Documentation

##  Overview

*Section 1 of this document identifies the scope of this document, the purpose for the software, and lists acronyms, definitions, and reference documents for this project. Section 2 provides an overview of the system written in terms of use cases. A use case diagram related the individual use cases, and use case scenarios describe the behavior throughout each of the use cases.*

# Use Case Scenarios

This section describes the use cases for this system. Included within this section are scenarios describing the behavior exhibited by each use case as well as the flows through the use case.

For each user case, there is a prioritization of that use case, with a lower number indicating higher priority being placed upon the given use case.

<Insert a Use Case Diagram Generated with EA in this location>

Figure Use case diagram.

## Provide Emergency Warning

### Actors

 Driver

### High Level Description

This use case scenario describes how a user would provide an audible warning to a person who is in the path of the robot. By providing an audible warning of the robot’s presence, the operator should be able to prevent a potential collision.

### Preconditions

Before this use case can be activated, the interface car software must have been started.

### Use Case Flow:

1. The operator detects something which is potentially in the way of the robot as it is being driven.
2. The operator activates the audible warning by pressing a key or clicking a button.
3. The vehicle emits an audible safety warning.
4. The operator determines that the audible warning is no longer necessary.
5. The operator deactivates the warning by releasing a key or pressing a button.
6. Safety warning stops

Alternate flows:

5-a-1. The audible warning has been active for more than 10 seconds. At this point, the control software will automatically disengage the audible warning.

# Specific Requirements

In this section, specific requirements for the system are stated. Each specific requirement is to receive a prioritization, where priority 1 indicates the highest priority requirement and priority 10 indicates the lowest priority requirement.

## User Interface Requirements

The user interface for controlling the system shall be implemented as a GUI. However, in addition to button presses, keyboard control may also be possible.

### General requirements

1. Splash Screen. Upon program startup, the RC Car Software shall display the name of the development team, the product name, the copyright year, and the members of the development team as a splash screen for 2 seconds before allowing normal operation of the RC car control software. (Priority 2)
2. All textual displays on the user interface shall have a minimum of 12 point font in order to be appropriately visible to the user. (Priority 4)
3. <Write This>…

### Ignition State

1. <Write These>…

### Grapple Control

1. <Write This>…

### Direction Control

1. <Write These>…
2. <Write This>…

### <Write The Rest Here>

## Functional Requirements

### Ignition

Ignition control shall control the behavior of the system in terms of an operating engine.

1. <Write These>… Note there may be some other headings.

## External Interfaces

## Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

### Performance

### Reliability

### Availability

### Security

### Maintainability

### Portability

### Safety

### Training-related Requirements

### Packaging Requirements

### Legal Requirements

* Copyright laws and license agreements must be respected for any third party software used in the creation of this system. (Priority 1)